FORTH USER

vol.2 no.1 1984

(formerly ACE USER).

Contents:

Coming soon: Simple robot for the Ace, Queues, N List, Alternative Case, Control Structures (all Ace); Spectrum FORTH, Book reviews, etc.

FORTH USER is edited and published by

JOHN L.NOYCE, Publisher, PO Box 450, Brighton BN1 8GR.

Subscription: £7 pa (£8 overseas). Free to current Jupiter Ace Users Club members.

JOHN L. NOYCE
Information Consultant
PO Box 450
BRIGHTON
BN1 8GR, U.K.

EDITORIAL

Hello again! Observant readers will have noticed the change of title. Formerly ACE USER, we are now widening our coverage to include FORTH USERs on other low-cost micros. Contributions welcome. The bulk of each issue will still be about the Ace, but coverage of other Forths will gradually increase. Comments please!

As most of you will know by now, Boldfield Ltd have taken over the bankrupt stocks of the Ace. The price is so cheap, that its worth buying a second Ace just as an 'insurance policy'.(Some-one else is selling Aces assembled from spare parts at an even cheaper price, but they seem to lack the top cover, and the manual. Details: Bob Black 0703-452407).

The Pacer rampack has ceased production. However Essex Microelectronics are still making their's. They also have a SOUNDBOARD, which we've seen and its very good, particularly as it has an I/O port as well. Boldfield have the Jupiter 16K rampacks, but none of the old 48Ks. We are still selling the ZX adaptor kit (soldering experience needed) so 48K is still possible. Details later in this issue of John Wike's colour board for the Ace (as we went to press we heard from one club member that it works and he seems pleased with it...).

The Microkey FORTH machine, which evolved from the REMSOFT project, is now available, details towards the end of this issue. Good value for the money, if you want that sort of hardware. The new Turtle dedicated LOGO computer at £70 looks interesting (see p10).

Club member Garry Knight has written a version of FORTH called G-FORTH, with graphics and games in mind. We hope to make this available on some of the newer machines soon (the Amstrad sounds interesting, as do the new Japanese MSX machines).

Several articles and reviews have had to be held over. See you next issue. Good reading! JN

CENTRONICS PRINTER INTERFACE FOR THE JUPITER ACE.

by David Prime.

The Centronics interface is an 8-bit parallel interface designed for high speed communications to a printer. The majority of the cheaper printers are fitted with this type of interface. Some printers are fitted with an RS232C interface which is a general purpose serial interface (see Ace User No.3). Since the Centronics interface is parallel, it is capable of much a higher data transfer rate. This is useful if your printer has a buffer since it allows you to to carry on computing whilst the printer is still printing.

This article describes how to implement the Centronics interface on a Jupiter Ace using the Essex Micro Electronics soundboard. It could easily be adapted to work with other ports provided you have 6 output bits and 1 input bit available.

The basic control signals required, apart from the character data are the strobe and busy signals. The strobe signal is issued from the computer to indicate to the printer that the data lines are to be read and a character printed. The busy signal is issued by the printer to tell the computer when it is not ready to accept data. The timing for these signals is given in figure 1. The Centronics interface also supports an acknowledge signal which the printer sends to indicate when the printer is ready to accept data. This interface uses the busy line since the acknowledge line only gives a pulse, and would therefore need an additional latch.

To implement a Centronics interface, you should therefore need 9 output bits (8 data bits and 1 strobe bit), and 1 input bit (busy). Unfortunately the soundboard only has the capability of 8 output and 8 input bits. A solution is to use 7 data bits, but this would only allow ascii characters to be printed and the graphics capabilities of many printers would be wasted. Another solution is to output the data as two four bit parts and provide a latch for one of the parts.
The latter solution is the one I decided to use.

THE HARDWARE.

The circuit diagram is shown in figure 2. To output a single character the following sequence must be performed (see figure 1 for timing diagram):

1) PB0 must be checked to see if the printers busy line is active. If it isn't active then a character can be sent to the printer.

2) The low 4 bits bf the character to be printed must be presented to PA0-PA3 together with PA5 (strobe) low. PA4, the latch for the 74LS75 must be given a high pulse. This will latch the 4 low bits of the character into the 74LS75.

3) The high 4 bits of the character to be printed must be presented to PA0-PA3 together with PA4 (latch) low and PA5 (strobe) low.

All 8 bits of the character have now presented to the interface cable via. the 74LS541 buffer. This data has to be stable before the strobe line is toggled.

4) PA5, the strobe line is now pulsed high. This gives a low pulse to the printer via. a transistor buffer.

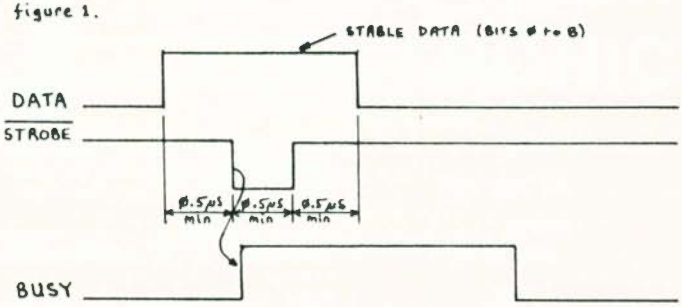
The printer should now set the busy line high and be reading the data. When the printer is ready, it will set the busy line low and we can send another character.

The interface was built on a piece of veroboard and the power for the interface was taken from the soundboard. Components are readily available from advertisers in most electronics magazines. The port connections are given in the soundboard manual. You should consult your printer manual for the connections to your printer. The pin numbers on the circuit diagram are for an Epson printer which uses a 36-way Amphenol plug (type 57-30360).

The software listings given provide words to initialise the ports and to print a character after checking the busy line. The routine given in Ace User No.3 (page 26), can be used to dump the contents of the screen if the word 'CH' is replaced by the word 'CHARPRINT' given.

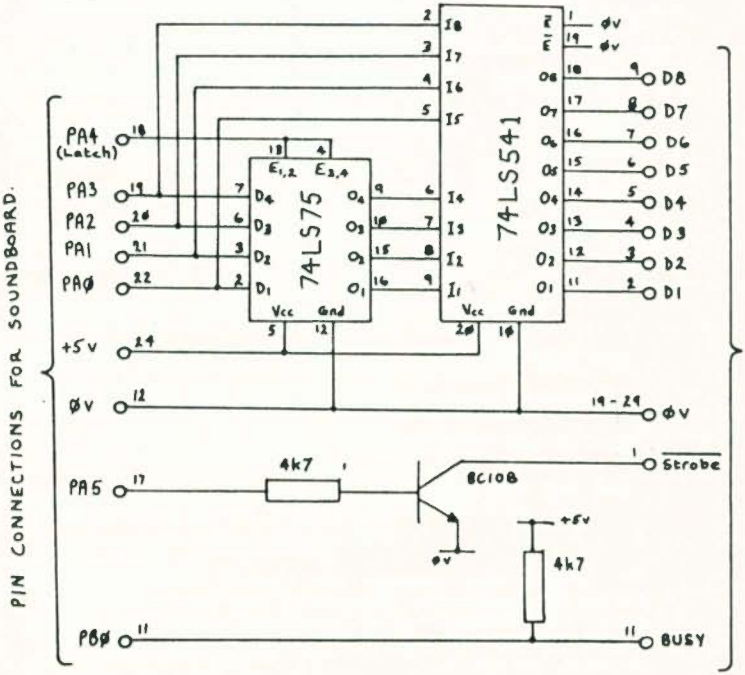
DATA TRANSFER SEQUENCE FOR A CENTRONICS PORT.

figure 1.



CENTRONICS PRINTER PORT FOR THE JUPITER ACE.

figure 2.



CENTRONICS INTERFACE DRIVING ROUTINES FOR THE JUPITER ACE E.M.E. SOUNDBOARD.

```
: PRINT-INIT        ( Initialise the ports.)
 7 253 OUT          ( Select soundchip control
                      register.)
 127 255 OUT        ( Set port A to output, port B to
;                      input.)

: BUSY              ( Wait for the busy line on the
                      centronics port to become low.)
 15 253 OUT         ( Select port B for further
                      operations.)
 BEGIN
  255 IN            ( Read port B to monitor the busy
                      line.)
  1 AND             ( Mask in the busy bit.)
  1-                ( Adjust value so that: 0=busy,
                      -1=not busy.)
 UNTIL              ( Test busy bit again if printer
;                      still busy.)

: CHARPRINT         ( c ... )
                    ( Output character 'c' to the
                      centronics port.)
 BUSY               ( Wait for printer to be ready
                      before sending the character.)
 14 253 OUT         ( Select port A for further
                      operations.)
 DUP                ( Make a copy of the character.)
 15 AND             ( Mask the low four bits of the
                      character.)
 DUP 16 OR          ( Copy the character and set the
                      latch bit high.)
 255 OUT 255 OUT    ( Output the low four bits of the
                      character. First with latch high,
                      then with latch low.)
 240 AND 16 /       ( Mask the high four bits of the
                      character and shift down four
                      bits.)
 DUP 255 OUT        ( Make copy and output the high
                      four bits of the character. The
                      data must be stable before the
                      strobe line is toggled.)
 DUP 32 OR          ( Make copy and set the strobe
                      bit high.)
 255 OUT 255 OUT    ( Output the high four bits of the
                      character. First with strobe
;                     high, then with strobe low.)
```

To print characters the word 'PRINTINIT' should first be executed. This will set the ports up. The word 'CHARPRINT' will send the number on top of the stack to the printer port, after checking that the printer is ready.

ESSEX MICRO ELECTRONICS

JUPITER ACE ACCESSORIES

SOUND & I/O BOARD: AVAILABLE NOW FOR JUST £30.

Three channel sound board with a 16 bit Input/Output port. Sound output is via. an internal loudspeaker or an external jack socket. Complete with 24 page instruction manual which also gives examples of software. We are also including details on how to make a centronics printer port using the soundboard.

FULLY CASED AND TESTED RAMPACKS.

16k Rampack	£25
32k Rampack	£35
16k to 32k upgrade	£13

EDGE CONNECTOR FOR ACE EXPANSION BUS & SOUNDBOARD I/O PORT: £4

SPECIAL OFFER:

SOUNDBOARD + 32k RAMPACK £59 (inc. P&P).

Send an SAE for details on the above products.

Please add £1 per order to cover P&P (except where stated).

MAIL ORDER ONLY. CHEQUES etc. to:

ESSEX MICRO ELECTRONICS,
4, HATCH ROAD,
BRENTWOOD,
ESSEX.
CM15 9PX

ROM EXPANSION FOR THE ACE

The question must now be asked, " Is it worth expanding the Ace? " We think that it is for two reasons, the first being that Ace forth is very good indeed, especially in its LIST, EDIT and REDEFINE facilities and the second is that it is very easy to write ROMs in forth that run directly on the Ace. This cannot be done on any other computer with comparable ease and it is certainly not possible to transfer BASIC programmes in this way. ROM expansion greatly increases the range and versatility of the Ace without demanding much increase in RAM. In many ways, building a new ROM repetoire for the Ace is like designing a new computer and is one of the most satisfying experiences for the hobyist.

Layout

There are three ways of adding to the Ace; directly via the edge connector; cable mounted via the edge connector and cable mounted via the Z80 socket. We have tried all three and prefer the second alternative where the extension board can be rested or fixed on top of the Ace. This gives a mechanically stable, compact and crash-free system. Expansion via the Z80 socket means mounting the CPU on the extension board and it opens the way for a major expansion in which the display and keyboard will be changed and improved. Meanwhile we have built a three-board system in which the edge connector is PCB mounted and cable-connected via DIL sockets to the main memory board. The eprom programming unit then fits via an inter PCB connector. ROM sockets are for 4K or 8K chips and the RAM extension is 4K relocatable static RAM, which can be placed in any 4K slot from 16K upwards.

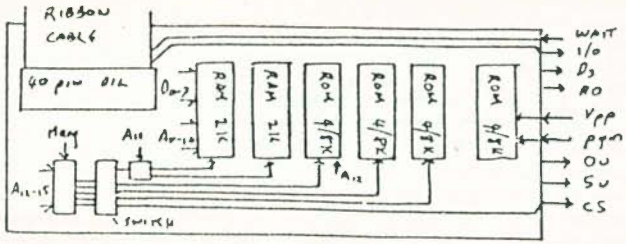
The Eprom writer

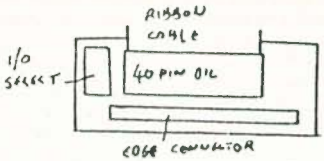
This is a new and very simple system in which WAIT is used to generate programming delays. It enables the ROM to be placed in normal memory and written by c! commands. A command " 8 N OUT " sets the system to write the Eprom and " 0 N OUT " restores it to normal read-only for the programming socket. The ROM can be written, verified and run on the same board. Software is easy to write, but system design requires some thought. We have a reasonably advanced Eprom which runs at 20-24K to enable the Ace to copy Eproms, write new ones, extend old ones, search vocabularies from the bottom upwards (cf. The forth method of searching from the top down) and to link or unlink any extension ROMs.

Forth Eproms.

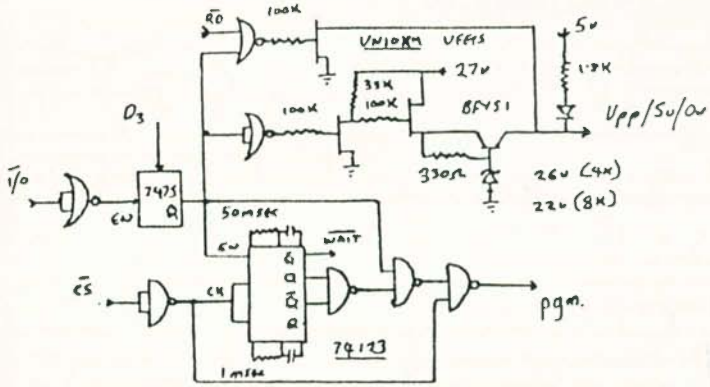
A Forth programme word sends a pointer to the linkage addresses of its component words and after each has been executed it returns to the next byte in the first parameter field. If a ROM is written in one memory location then these jumps to and from other ROMs only work for that location and are invalid for other locations. The structure of FORTH is so simple and elegant that once the linkage addresses are correctly handled there are no other problems in the use of ROMs!

ROM/RAM extension board with one Eprom programming socket





Ace adaptor



Eprom programming adaptor

note: on some Aces the output to WAIT may need an OPEN COLLECTOR (info from the authors by phone as we went to press).

When a programme is written in RAM above HERE the words are linked down to FORTH and the linkage address of the top word is stored in address 15436, the place where the Ace begins its vocabulary searches. If the same programme is transferred directly to ROM, it also links down and is valid. When the Ace is powered-up, however, it looks for a linkage address stored in 15436 and finds 15433 the address of FORTH (Which it transfers from addresses 309-10 in Ace ROM A as part of the setting-up proceedure). The new ROM vocabulary can only be found if its top linkage address is either poked into 15436, or implanted into bytes 309-10.

In order to write programmes for a ROM, the ROM CS must be swapped with that of an equivalent block of RAM. In most cases this RAM will then be separated from the on-board RAM and RAMTOP will be 16384. Vocabulary must be written entirely within isolated extension RAM and to do this its top address must be reassigned as RAMTOP (ie n 15384 !) and its bottom address as HERE (ie m 15415 !). This gives an obligatory ERROR 2 which must be ignored. Words can now be written and new active vocabulary compiled. When it is finished find HERE and obtain the top link address by 15436 ! . and then define a word : x ; which acts to complete the vocabulary below it by filling in the last length field value. Copy the RAM programme into ROM between the two HEREs. Relocate the memory blocks in the normal configuration and activate the ROM vocabulary by poking its link address into 15436.

A.J. and P.D.Lawrence. March 1984.

Department of Cell Biology
University of Glasgow
Glasgow G12 8QQ

TEL: 041 339 8855 ext.7196

COULD BE INTERESTING DEPT:

An Anglo-Israeli microcomputer dedicated to the educational language Logo will cost under £70.

The Turtle Computer, or TC 10, named after the physical 'cursor' employed for external demonstration of a Logo program in action, is being distributed by London-based educational toys firm Mindstrain.

Michael Elliott, Mindstrain's managing director, said that the TC 10 has been developed by academics at the Jerusalem University to teach children. It is an 8086-based micro which can only be programmed in Logo, but which is cheap because it is dedicated to the single task of interpreting Logo applications.

Schools supplier EG Arnold "is considering taking on the TC 10 as one of its Offspring: Early Learning products", said Elliott. He is looking for dealers in the education field to take on the machine when first models are produced by Thorn-EMI in June.

The TC 10's graphics are exceptionally good, and it has an RS232 output to guide one of the Turtle devices if required. It is housed in a robust red metal casting "which kids can't harm", added Elliott.

Source: Microscope (Br. computer trade paper)

EDITING AND SAVING SCREEN DISPLAYS ON THE JUPITER ACE

M. Greenwood

This FORTH word SCRN* has three functions. The first is a screen editor to set up a display. Graphics and inverse controls are as normal, the mode being indicated by the form of the cursor, which in this case winks over the character in the cursor position in the display. The cursor can be moved in the usual four directions plus diagonal down left (shift 2) or right (shift Ø), and cursor direction of movement is repeated on each character insertion until changed by a direction control. Shift 1 clears the screen and <enter> finishes the edit, so don't loiter on <enter> after "SCRN Name" (which is the syntax for this word).

The second function is to store the screen that has been set up in the dictionary word "Name" (as chosen) in a compact form that can be described as a transparent overlay bearing opaque characters. Most systems of display store the 736 bytes 9216-9952. This word is suitable for the typical display which contains a fair proportion of spaces, and it stores each sequence of spaces as two bytes viz. 'space' and 'no. of spaces'. Typically a screen will be stored in 2-3ØØbytes, economising up to .5k of memory on a screen. The drawback is that this system doesn't work for displays containing a sequence of more than 255 spaces viz. about six adjacent blank lines. I get round this by putting borders around displays.

The third function is executed whenever the word 'Name' is called, e.g. by a word calling a number of similarly generated words. In execution the display is re-created as an overlay. The stored spaces are not printed - the word, instead, skips over that number of screen locations before printing the next character, so any previous display is not obliterated by the blank parts of the new one. This is useful in certain animation situations - unwanted parts of mid- or backgrounds can be deleted, e.g. by using AT and SPACES words. Memory can be economised by REDEFINEing the editor SCR (which is the bulkiest part at 45Ø bytes) when no longer required. The remainder occupies about 3ØØbytes.

```
(SCRN* requires these to run on)
Ø  VARIABLE  CU  (cursor char.)
Ø  VARIABLE  PR  (pointer)
Ø  VARIABLE  C+  (cur. direction)
Ø  VARIABLE  SS  (spaces count)
Ø  VARIABLE  CH  (char. store)
:  +!  (as per manual p. 44) ;
:  WINK  (code code addr WINK, winks two chars. at address)
DUP  ROT  SWAP  3  1  DO  C!
3ØØ  Ø  DO  LOOP  LOOP  ;
```

```
:  SCR  (a screen editor)
(leave time to quit <enter>)
INVIS  3ØØ  Ø  DO  LOOP
(put char. at pointer in store)
PR  @  C@  CH  !
(wink char. at pointer against cursor char., key and repeat)
BEGIN  CH  @  CU  @  PR  @
WINK  INKEY  ?DUP  UNTIL
CH  @  CU  @  PR  @  WINK
(on <enter>, finish)
DUP  13  =  IF  DROP  EXIT  ELSE
(if control key pressed)
DUP  11  <  IF  DUP
(on shift 1 clear screen)
DUP  10  =  IF  CLS  DROP  Ø  THEN
(on shift 4 for 'inverse')
DUP  8  =  IF  CU  DUP  @
(toggle 128s bit of code in CU)
128  XOR  SWAP  !  DROP  Ø  THEN
(if cursor direction 'reverse')
DUP  1  =  IF  DROP  -1  THEN
(on shift 8 for 'graphics')
DUP  4  =  IF  CU  DUP  @
(toggle 8s bit of code in CU)
8  XOR  SWAP  !  DROP  Ø  THEN
(cursor direction controls-)
(leave cursor operand on stack)
DUP  2  =  IF  DROP  31  THEN
DUP  3  =  IF  DROP  1  THEN
DUP  9  =  IF  DROP  32  THEN
DUP  7  =  IF  DROP  -32  THEN
DUP  5  =  IF  DROP  33  THEN
(operate and store last cur. control)
DUP  C+  !  PR  +!  ELSE
(not a control char., to print; fetch 8s bit from CU and adjust
 product of INKEY for 'graphics')
CU  @  DUP  8  AND  Ø=  3  ROLL
SWAP  IF  32  MOD  THEN  SWAP
(fetch 128s bit from CU and adjust for 'inverse')
```

```
128  AND  Ø=  Ø=  128  *  +  DUP
(print at current pointer, and store)
PR  @  C!  CH  !
(move cursor on as last directed-)
(unless mode just changed)
C+  @  PR  +!  THEN  THEN
(if pointer now off screen, reset)
PR  @  DUP  9216  〈SWAP  9952〉
OR  IF  9216  PR  !  THEN
(recur until finish on <enter>)
SCR  ; (REDEFINE  SCR  when no longer required & regain 450
 bytes)
DEFINER  SCRN*  (main word)
SCR  (edit screen until finished)
79  CU  !  (start cursor char.)
42  9952  C!  (mark end of screen)
Ø  SS  !  (set spaces count to 0)
9952  9216  DO  (visit each screen addr)
I  C@  DUP  32  =  IF  (a space)
1  SS  +!  (add to running total)
DROP  ELSE  (a character)
SS  @  Ø〉  IF  (preceeded by spaces)
32  C,  SS  @  C,  Ø  SS  !
(enclose in dict. 'space' and 'number of spaces', reset
 counter)
THEN  C,  THEN  LOOP  (enclose char.)
DOES〉 (runtime action for 'name')
PR  !  (pointer to start of data)
9952  9216  DO  (for each scr. addr.)
PR  @  C@  (fetch char. from data)
1  PR  +!  (move pointer on)
DUP  32  =  IF  (a space)
DROP  PR  @  C@  1  PR  +!
(discard and fetch next from data)
ELSE  I  C!  1  THEN  (print it and jump either the number
 of spaces or to next which might be a char.)  +LOOP  ;
```

LETTERS section (EH=Edward Hattersley
JN=John Noyce)

Dear JAUC,

Having put up with the Ace's keypads for 10 months I now want to fit a full-travel keyboard. The choice seems to be between the Fuller FD42/FDS and the Ricoll Electronics RIKB1. Have you any suggestions as to which is the easiest to adapt for the Ace? I am not too enthusiastic about soldering wires directly to the Ace PCB, so I want to interface the keyboard via the Ace's edge connector. Would it be possible to simply solder the keyboard matrix wires to address lines A8-A15 & data lines D0-D4 on the edge connector, or would the keyboard emvlation circuit designed by Doug Bollen be necessary? (Mentioned in Ace User 4). If it is possible to connect up a ZX-81-type full travel keyboard to A8-A15 & D0-D4, what is the point of such an emulation circuit? Any information/hints about fitting a proper keyboard to the Ace would be greatly appreciated.

I include a SAE for your reply. Keep up the good work!

Yours faithfully,

Jonathan Hardwick.

I would advise you to use the Ricoll KBD, as Fuller's seem to be causing some problems (see other letters). Doug Bollen's device, or similar, is definitely necessary, as it isolates the KBD from the data lines when the keyboard is not being accessed (as in the ACE itself-see circuit diagram in Ace User 3). If you connect a KBD directly to the address and data lines, the machine will crash instantly every time you press a key.

N.B. A ZX-81KBD will require some alteration before fitting to the ACE due to a slightly different key layout-see page 3, Ace User 3.

E.H.

Dear John,

Please find my subscription for this year. Keep up the "Ace User", I find it most useful.

I am most interested in your comments on Messrs. Fuller. You may be interested to learn that I ordered a keyboard from Fuller's last September and despite frequent letters got no replies. Finally I received a keyboard for a Spectrum in late December! Since, despite further letters I have still no satisfaction.

You may publish this!

Yours sincerely,
John Stone, Tiverton.

(letter date: 31/1/84. Yes, Fuller's have been sailing close to the wind. For anyone else in the same boat, write a firm legally-sounding letter of the "if you do not deliver in x days, I'll ...." -J.N.)

There is a disadvantage to using strings as shown in the manual because you can't LIST or EDIT them. Here is my solution.
Define a word,

```
:  DATA
 I  2+  DUP  2+  SWAP
 C@
;
```

Then you can define strings like this:

```
:  FREDSADDRESS
 DATA  (23, Flightpath Lane, Heathrow)
;
```

with the string inside a normal ACE comment. Strings defined in this way work exactly the same as using DEFINER STRING. Any other uses for ACE comments along this line?

Many people seem to have overlooked the fact that you can print in the input buffer by Flipping bit 3 of FLAGS. This can be used when saving within a program, eg using the method above.

```
:  FLIPFLAG
 15422  C@  8  XOR  15422  C!
;
:  .BUF
 FLIPFLAG  I  2+  DUP  2+
 SWAP  C@  TYPE  FLIPFLAG
;
:  DEMO
 .BUF  (Program-name)
 SAVE
;
```

Then DEMO will save the program. .BUF prints the comment in the input buffer. Be careful when using FLIPFLAG!

I hope this will further the use of some people's ACE's.

Colin Dooley.

Dear Sir,

I saw your advertisement for the Jupiter Ace User's Club in the November issue of Computing Today, and I would be grateful if you would send me details as soon as possible (SAE enclosed).

As a hardware enthusiast, I have been developing a more complete "system" around the Ace hardware. This system has the following features:

I Modified bus system (fully buttered)
II Full memory expansion (48K dynamic)
III Full QWERTY keyboard modification
IV Sound board
V Speech processor/synthesizer
VI Hi-res graphics

I am also considering an assortment of analogue and digital interface boards, page mode memory system, Eprom board, etc.

At the present time boards I and II are completed, I have final circuit diagrams for IV and V and I have started work on VI using the "Thompson" series of VDP chips to produce a 512 x 512 colour system. Board III awaits the arrival of an ex-equipment keyboard that I have ordered.

The reasons for telling you this are as follows, firstly I assume that you publish a newsletter and I would be grateful if you would pass the information on to other members as I am willing to release a limited number of circuit diagrams (limited because I hope to have the system published!!!). And secondly I would be interested in hearing from anyone working on similar projects so that we can achieve a measure of standardisation over a complete system bus, card sizes, etc., etc. Finally I really could use the help of a rather more capable programmer than myself as the system would benefit enormously from a more complete software/hardware combination than I could come up with on my own.

If the club feels that it is interested in this system, I would be delighted to submit my suggestions for the bus system to the club immediately, so that others working on hardware can criticize/complain with the object of providing the "perfect" expansion bus for the machine. Once the bus is perfected (assuming any problems are found) it would be a simple matter to interface all hardware to this "club bus" on a standard sized board (?) in a rack of some sort.

Once again, any suggestions, criticism or help from club members will be most gratefully received.

Yours faithfully,

Stephen Jackson,
51 Wadsley Lane,
Sheffield S6 4EA,
S Yorks.

George Baker, 37 Ormerod Rd, Burnley, Lancs, would like to get in touch with nearby Ace users. (phone call as we went to press)

With reference to M. Finley: Forth Control Structures, in ACE USER 3 I must point out that the word BACK is NOT equivalent to the FIG-FORTH standard BACK. Mr. Finley CASE structure in ACE USER number 3, is very well written, and his use of COMPILER...RUNS› shows the clever way to program.

I hope you have read the excellent article in SOFT: Pascal in Forth, where Pascal extensions are added to Forth i.e. FOR := WRITELN etc. I have modified the program, and it is up and running after some conversions to the ACE. I include a few small Forth words below.

```
:  BLOT  (wipes out COMPILER reference)
        HERE 2- 15415 !
;
Ø  COMPILER (COMPILE)  FIND  BLOT  ,  RUNS›  ;
:  COMPILE  15422  C@  64  AND  IF  (? compiling)
   R›  DUP  2+  )R  @  ,  ELSE  (display error message  Ø)
   Ø  15421  C!  ABORT THEN  ;
```

Maxwell Nairn Andrews

Dear JAUC,

I enclose a cheque to cover my subscription during 1984 and a keypads kit (I can't stand the Ace keyboard much longer, so please deliver as soon as possible!). Apart from Doug Bollen's plug-in interface mentioned in issues 3 & 4, are there any ZX-81 keyboards which have proved easy to adapt for Ace's (soldering wires direct to the Ace pc6), or has any manufacturer taken Fuller's place in producing a special Ace keyboard?

Loading hint: my Ferguson 3TO7 tape recorder only gives error-free loading when 6 ft away from the tv set and shielded by a table-top. Jupiter Cantab said they too had had tv problems with Ferguson tape recorders.

The cursor character on the Ace is ascii 151, the inverse of ascii 23. Thus, to redefine it, set character 23 to the inverse of the new character using GR from the manual.
e.g. to redefine it as an underline ("_") character:
255 255 255 255 255 255 255 0 23 gr
The catch is that character 23 is used by the PLOT routine, so programs such as DRAW from the manual will give some pretty weird displays when used with a redefined cursor!

Keep up the good work,

Jonathan Hardwick
Stratford-on-Avon

Has anyone discovered what error code 16 means?

P.J. Guy, Horsham

Or error code 66? JN

Thank you for Ace User 4. I have written to ask if you are aware of anyone -or company- who are willing to repair a Jupiter Ace.

With regard to Iestyn Walters' question and your comments about lack of a sync line the following comes directly from Electronics + Computing Monthly (September).

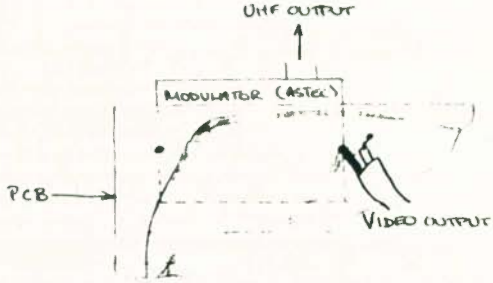
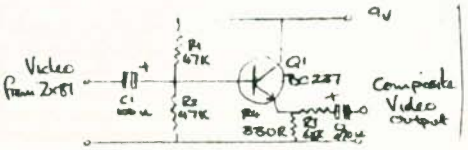
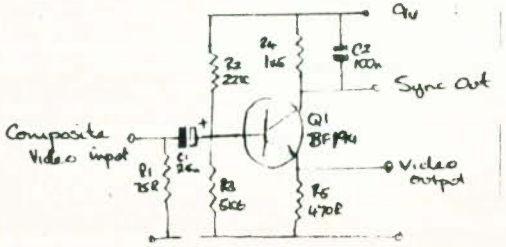


Diagram showing the points on the ZX81's PCB to which the direct video connections are made.

The cable recommended by ECM is a lightweight screened audio cable with the screen connected to the righthand lug and the center core to the PCB as shown, which corresponds to one of the wire connections to the modulator (this gives just two possible points). In most cases the output will not be of the Standard 75R impedence used in Video circuits; the following circuit was given to give 75R impedence (Is the output on the edge connector at 75R impedence?).



The circuit diagram of the buffer stage that may be necessary in order to convert the high Impedence Video output of the ZX81 to the standard 75R level. It can be run from ZX81 AV rail.



The circuit if a separate sync signal is required (as with some RGB monitors). The circuit is driven from a composite video input.

(P46-47 ECM Sept)

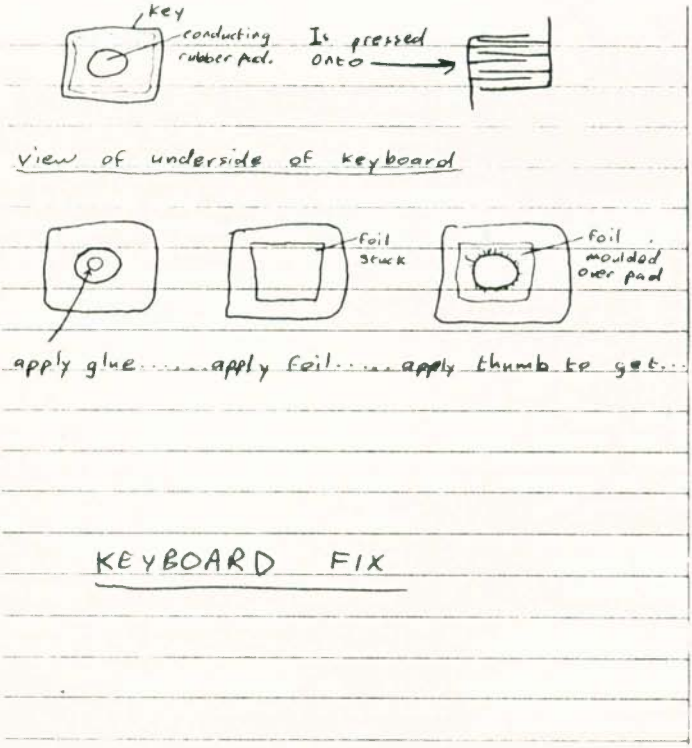
3 Tavora Street, Pinderfields Rd,
Wakefield West Yorkshire WF1 3NP.
D.Tomlinson,

Dear John,

After a lot of experiment I have come up with the following easy method of improving the ACE keyboard by about 1 x 10 100%. The ACE keyboard works by a conductive rubber pad being pressed onto tracks on the circuit board thus making contact across them. This method is rather insensitive as the rubber is not a very good conductor. Glueing small squares of ordinary kitchen foil onto the pads (see diagram) produced an excellent result and the keys now work with a very light prod. I hope this will help other ACE users, I know my keyboard had 3 keys which only worked by vigourous massaging of the key, this drove me mad as one was the shift key (try editing with that!).
R. Mayell wanted a machine code Random number generator. Here's mine, it uses the useful fact that when a call to a subroutine is made, HL points to the start of the code. And thanks to M. Finley for a novel use of compiler!

Colin Dooley

(This is the principle used by Lars Arell in his keypad kit (£2 to club members...) JN.)



	RND
Assembler	HEX
PUSH IX	DD E5
PUSH HL	E5
POP IX	DD E1
LD H,(IX+23H)	DD 66 23
LD L,(IX+24H)	DD 6E 24
LD D,L	55
LD E,29H	1E 29
ADD HL,DE	19
LD (IX+23H),H	DD 74 23
LD (IX+24H),L	DD 75 24
LD A,R	ED 5F
XOR L	AD
LD L,A	6F
XOR H	AC
LD H,A	67
EX DE,HL	EB
RES 7,D	CB BA
RST 10H	D7
POP IX	DD E1
JP (IY)	FD E9
DEFW SEED	00 00

Any members in the Surrey area who feel like having a chat about Forth and/or ACE (preferably not games) can phone me at REIGATE (Surrey) 43435.

A.W. Ambrose,
8 Knights Court,
Reigate Road, Reigate
Surrey RH2 0QY.

Please tell me whether the following is happening to just my Ace and no one elses'. When C!'ing bytes into the character memory, sometimes bits are corrupted. This is very noticeable with the software I got with my ZX printer-Ace adapter and is very annoying.

John Kennedy, Bangor, NI.

10.45 *Stereo*
Daily Service
God or Mammon?
NEM, p 25; Forth in thy name (BBC HB 406); Jesu, lead my footsteps ever (Christmas Oratorio); Mark 12, vv 1-17; Christ is our corner-stone (BBC HB 258)

(← RADIO TIMES)

Dear Sir,

I would be grateful if you could answer my question: I have noticed, recently, the Alphacom printer for ZX81 and Spectrum. Have you tried this product? I assume that it should be compatible with the software from tape 11. Is this so? I hope you will be able to help me. Perhaps you could include your answers with the tape 20 which is still outstanding.

As regards the Fuller FDS keyboard, I have sent Fuller a letter from my Solicitor, threatening legal action if the matter is not resolved immediately. I will write to you in a few weeks time, commenting on whether the product in question is worth all the aggravation it has so far caused me.

I thank you in advance, and I hope that the Users' club runs for a long time yet.

Yours faithfully,

J.P. Lechocki.

Alphacom printer should work, but we haven't tried it... Do let us know how you get on! J.N.

Works very well with Goldfield Printer Software.

WORD DATA (Ram only)

A.W. Ambrose

Issue 2 pages 14/15 were found to be interesting by using FIND WORD READ. My version uses READ WORD and prints all the byte contents that follow the word, i.e. READ COUNT (DUP 1+ SWAP C@)

```
  READ COUNT
  17  Ø  157  63  5  195  14
  1Ø7  8  9  14  133  8  15Ø
   8  182  4
  OK
: READ
  FIND  CR  5-  DUP  DUP  C@  +  SWAP
  DO  I  C@  .  LOOP  CR  ;
```

If you have HEX (16 BASE C!)

Then enter HEX READ COUNT

ADDING COLOUR TO THE ACE

Black and white is boring, so add a little colour (or a lot, if you want) to your computer's life with this project from John Wike. Additional material by Phil Walker.

The two facilities that really enhance the output of a microcomputer are sound and colour. The Jupiter Ace already has sound of sorts and this project provides a means of adding colour to its monochrome display. Eight colours including white and black are available, any of which may be selected as foreground (ink) or background (paper). The circuit is active from switch-on and requires no special software for monochrome operation. Thus programs may be listed, edited and run without the need to keep swapping over aerial leads or operating systems. The board uses the smaller edge connector and contains a RAM (addressed in parallel with the Ace video RAM) to hold the colour information for each of the 768 locations on the screen.

EDITORIAL COMMENT

It may seem a little strange to publish a project for the Jupiter Ace several months after the manufacturer has gone into liquidation. However, there will be a number of readers around who have Aces, and, moreover, we think that there is a lot to be learned from the techniques and systems described here. We think you will agree with us once you've read this article.

Attributes

Each character written on the screen will have associated with it the current attribute describing its ink and paper colours. To print anything with different colours the current attribute must first be updated. This is easy to implement and software for doing so is given later. The default colours at switch-on are green ink and black paper. The attribute number is obtained by adding the ink value to the paper value and adding 128 (80h).

Colour	Ink	Paper
Black	0	0
Blue	1	16 (10h)
Red	2	32 (20h)
Purple	3	48 (30h)
Green	4	64 (40h)
Cyan	5	80 (50h)
Yellow	6	96 (60h)
White	7	112 (70h)

About The Ace

The Jupiter Ace video display consists of 24 rows of 32 characters, each of which is selected by storing the required code at the appropriate location in the video RAM. There are 128 character shapes (plus their inverses) available and these are re-definable by the user.

At the rear of the computer

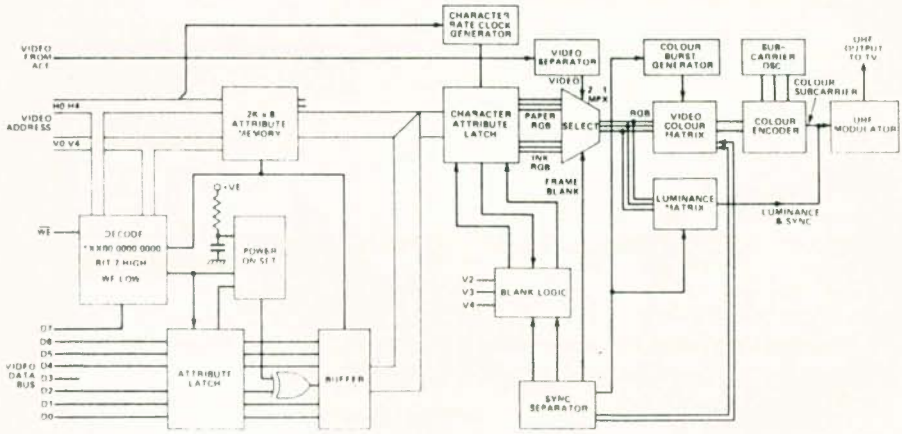
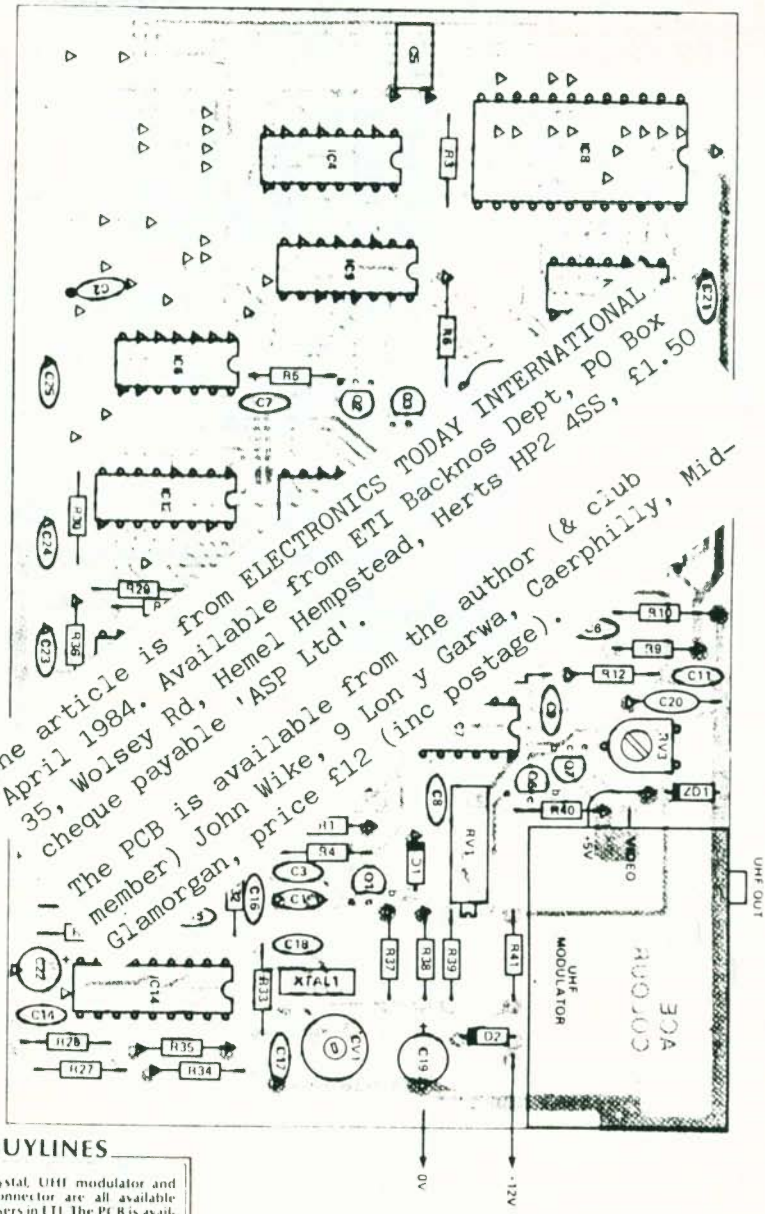


Fig. 1 Block diagram of the colour board.



BUYLINES

The IC's, crystal, UHF modulator and the edge connector are all available from advertisers in ETI. The PCB is available from the author at 9 Lon y Garwa, Caerphilly, Mid Glamorgan for £12 including postage.

△ SOLDERED ON BOTH SIDES (COMPONENTS AND LINKS)

● SOLDERED TO TOP FOIL ONLY, WITH NO LEAD THROUGH

PERSONAL COMPUTER NEWS
PCN PRO-TEST HARDWARE
April 14, 1984

Forth feature

Ted Ball looks at the prospects for a new Forth system.

Nearly a year ago, PCN readers were asked to contribute suggestions for the ideal micro. The 1,700 responses resulted in the Microkey.

In theory a computer is a general-purpose machine that can, with suitable programming and connections to the outside world, be adapted to perform any task we require. In practice each model comes equipped with a range of programs and interfaces that give it an identity and point to specific areas of application.

The Microkey 4500 does not fit into any of the usual categories. The hardware and price suggest a business system, but the machine lacks a standard operating system that would allow it to run widely available business applications programs. It resembles a home computer in having a programming language and its own operating system in ROM, but the programming language is Forth instead of the usual Basic.

The clue to the Microkey is, of course, the Forth language. Forth was developed originally for programming applications involving the control of scientific and industrial equipment, and the Microkey is ideally suited in its hardware and software for such applications, and also for use in technical education.

First impressions

The system is intended for experienced computer users, and although the manual includes sections on setting up the system and getting started, a novice is likely to have difficulty with this.

The system consists of the computer cabinet (which may include up to two disk drives), a keyboard and a monitor, which have to be plugged together. Connecting up the system is quite straightforward if you have a little technical knowledge, otherwise the terse technical labelling on the diagram of the rear panel connections could be confusing.

Once you have the system set up and switched on you have to boot the system disk following step by step instructions in the manual. The machine is then ready for you to start programming.

Documentation

The documentation consists of a reference manual and an introductory textbook on Forth.

The textbook is *The Complete Forth* by Alan Winfield, published by Sigma Technical Press. Although not written for the machine, it deals with the same standard Forth 79 language that is implemented on the Microkey. The book covers all the features of Forth systematically, with examples and exercises to try out. It is a first-class introduction to programming in Forth, and has been widely recommended both for self-study and class use.

The reference manual gives technical details of the hardware and software and includes everything you need to make full use of the Microkey 4500 system. It covers the program editors, the Forth vocabulary (with extensions to Forth 79 listed separately), the 6502 assembler in Forth, the hardware specification (with details of all the interface connectors and pinouts but no circuit diagrams), the system monitor commands and the use of operating system routines.

The manual is purely a reference document and you need considerable technical knowledge to understand it fully.

Keyboard

The keyboard is an IBM type, and is the only thing about the Microkey that I really dislike. There has been so much criticism of the IBM keyboard that I don't need to say much here, but my main objections are the non-standard layout and the too springy feel of the keys.

Display

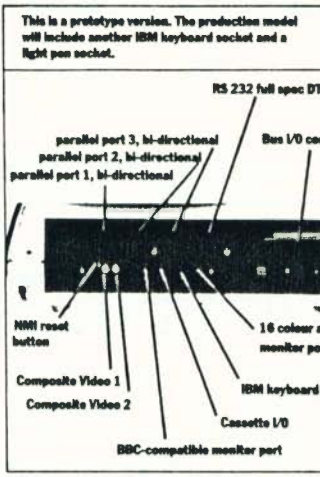
The Microkey 4500 has two high-resolution graphics display modes. In colour it gives 640 by 200 pixels with eight colours available (which can be extended to 16 colours by fitting an additional colour card), and in monochrome, 1280 by 200 pixels. There are no separate text and graphics modes; text and graphics can always be mixed on the screen.

The resolution available is more than can be handled by a normal TV, so no UHF output is provided. Microkey Ltd recommend the Cotron Sword 16 colour analog monitor, but the Microkey 4500 also has video outputs for an RGB monitor and two monochrome monitors.

Hardware

The Microkey 4500 is based on a 6502 microprocessor running at 1.843 MHz, with 128K RAM and 32K ROM. It has a large number of interfaces, two parallel input/output ports, 7-line bi-directional RS232, and connectors for four disk drives (which may be any mixture of Sony 3½in, Epson 3½in, and standard 5¼in), and a system bus compatible with the Apple II bus.

The machine has been designed to accept more modern processors than the 6502 in order to ensure its long term usefulness. The additional processors allowed include the 6809, (although no firmware is available for this yet) and the Western Digital 16-bit microprocessors W65SC816 and W65SC802. These Western Digital processors can operate as an 8-bit 6502 (so no new firmware is required) and also as full 16-bit processors with an enhanced instruction set. The Microkey's RAM can be expanded to 512K if the W65SC816 is fitted.



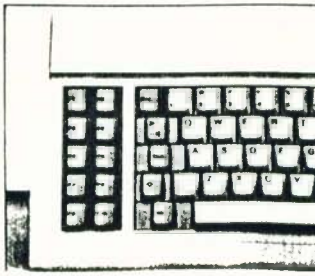
This is a prototype version. The production model will include another IBM keyboard socket and a light pen socket.

Software

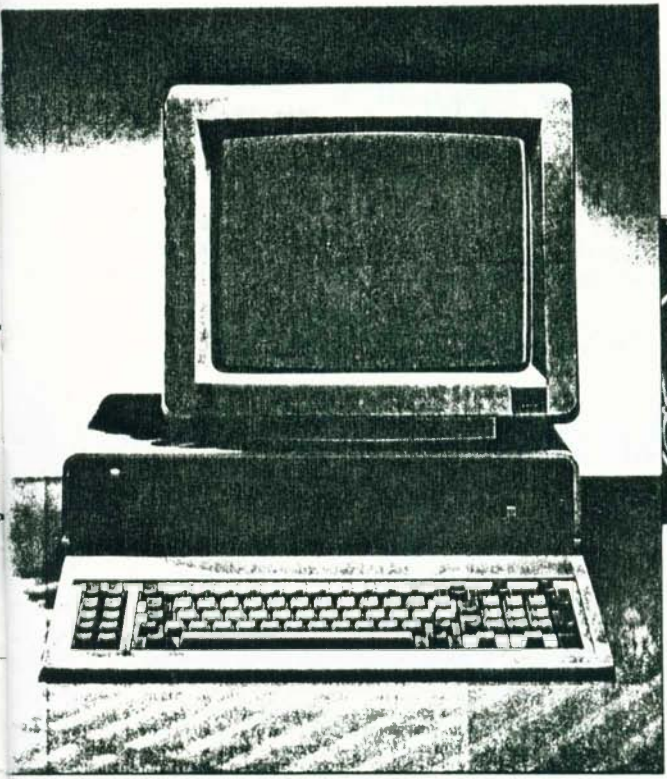
The major part of the software is the Forth programming language, which is held partly in ROM and partly on disk. It is standard Forth 79 with a few extensions. The extensions consist of some system dependent words, for interfacing with the system monitor and handling the graphics, and some Fig Forth words that were omitted from the Forth standard.

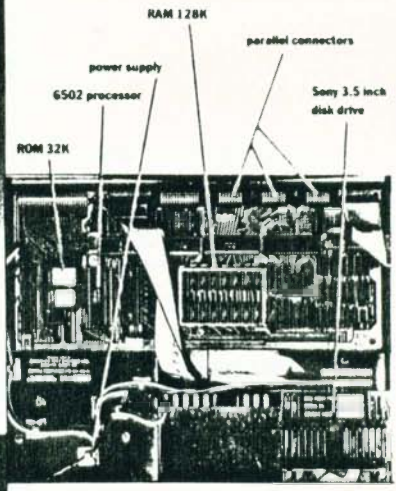
Other software includes a system monitor in ROM which allows various input and output operations independently of Forth, an operating system ROM with the input and output routines documented so they can be called from user programs, and a 6502 assembler in Forth.

The only software available for the Microkey so far is what is included with the machine. However, there are many programming utilities in Forth available as source listings or on disk for other systems, and it is likely that much of this software will become available for the Microkey once the machine is in production.



Paul Wilson





Utilities

One of the advantages of a standard programming language is you don't have to re-learn it or rewrite your programs every time you move to a new machine. I was able to use the Microkey quite easily after a few minutes' study of the section of the manual that describes the non-standard and machine dependent features of Microkey Forth.

The Microkey includes the usual line-oriented editor that comes with Forth, but this is difficult and tedious to use, requiring separate commands to select a line of program, position in the line and for insertion and deletion of text. However, there is also a screen editor which is much easier to use, allowing you to move the cursor to the place you want to edit and then insert or delete directly.

Although the Microkey system has a cassette interface to allow it to be used without disk drives it is very difficult to use with cassette only. The system monitor allows you to save and load blocks of memory on cassette and it is possible (though not documented) to save and load programs in this way. However, without a disk drive it is not possible to edit programs without retyping everything.

The review machine included one Sony 3½in drive, which made it possible to use the machine in the way it was intended. The main problem with only one disk drive is the difficulty of making a backup copy of a disk. It can be done, but involves frequent changing of disks. The disk copy utility in the Microkey only works when you have two disk drives.

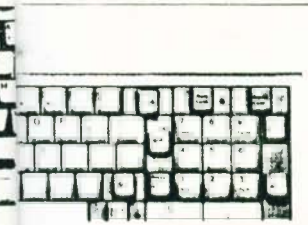
Forth is used a lot in control applications because it is almost as fast as machine code but much easier to write than assembly language. The speed of the Microkey is demonstrated by the software that allows two users with separate keyboards and monitors to operate the machine at the same time.

Support

Microkey Ltd intends to provide full support for the machine including the new processors mentioned earlier, extra hardware including Centronics and modem interfaces and additional software. The company will also be issuing its own 'Seal of Approval' for hardware and software provided by outside suppliers.

The Microkey 4500 is aimed at scientific and industrial control applications and at education in these fields. The built-in Forth and the many interfaces make it well suited for these uses, and with the total cost being comparable to that of any similar hardware. The robust construction and forward-looking design make the machine good value for money.

Although the Microkey 4500 is not meant for the ordinary home user, I would expect many Forth enthusiasts to buy a Microkey system for their personal use.



SPECIFICATIONS

Price Microkey 4500: £748. Cotron Sword 16-Colour Monitor: £402. Sony 3½in disk drives: £241 each.
Processor 6502 (which may be upgraded to 6809, W65SC816 or W65SC802)
RAM 128K.
ROM 32K.
Text screen 80 by 25 or 160 by 25.
Graphics screen 640 by 200 in 8 colours or 1280 by 200 monochrome. Analog, RGB, and 2 composite video outputs.
Keyboard IBM types, 87 keys, 10 function keys and numeric keypad.
Storage Up to 4 disk drives, single or double density, which may be any combination of Sony 3½in, Epson 3½in, and 5¼in.
Interfaces RS232, 2 parallel user ports, 2 IBM-type keyboard interfaces, Apple compatible cassette interface, Apple compatible system bus.
Language Forth 79 with extensions.
Distributor Microkey Ltd, 98a St. James's Street, Brighton, Sussex BN2 1TP. (0273) 672911.

Microkey 4500 Specification.

1. Microprocessor :-
Choice of 6502, 65C112, 6809E, W65SC816, W65SC802 (presently supported).
Speed of 1.843 MHz, 542 nS cycle time, is standard (DMA from expansion bus is not possible with 6502).

2. RAM :-
128K bytes of dynamic RAM, in 2 X 64K byte banks, is standard. (Expansion to 512K byte is possible with W65SC816). Refresh is done automatically by video controller. RAM is used by video in first half of each cycle, and by micro in the second.

3. EPROM :-
32 K bytes using 2 X 27128, 16 K bytes using 2 X 2764.
EPROM 1 start address=$8000.
EPROM 2 start address=$C000.
EPROMS can be completly switched out, or replaced by external plug-in EPROMS.
Note that a write to any of the EPROM address space will always write to the corresponding RAM location irrespective of EPROM configuration.

4. Serial interface :-
TX, RX, CTS, DCD, DSR, DTR, RTS are provided, at RS232 Voltage levels (-12V to +12V).
Connector is 25 way D type plug. 16 software programmable Baud rates, up to 19200, are possible. A 6551 is used, register addresses are :
$FF00 - RX Data/TX Data
$FF01 - Status
$FF02 - Command
$FF03 - Control

5. User port 1 :
An 8 bit bidirectional port, with 2 handshake lines, is provided. TTL Voltage levels are used (logic 0 0.8V, logic 1 2.4V).
Connector is 15 way D type plug, 0 and 5V are also available on this connector.
Port A of 6522 is used, register addresses are :
$FF11 - Port A Data register
$FF13 - Port A Data direction register.
$FF1C - PCR (handshake control).

6. User port 2 : Same as above.
Port B of 6522 is used, register addresses are :
$FF10 - Port B Data register
$FF12 - Port B Data direction register
$FF1C - PCR (handshake control)

7. Keyboard interface :
There are 2 IBM type keyboard interfaces provided. Signals are clock, data, 0V, 5V, Reset , and are at TTL Voltage levels.
Connectors used are 180 degree 5 pin DIN sockets

8. Cassette interface.
An Apple compatible cassette interface is provided on a 5 pin DIN socket.

9. Expansion port.
All bus control and power supplies are provided on a 50 way IDC plug.

10. Screen 1 video output.
Screen 1 output is composite video monochrome, resolution is 640 X 200 pixels, 1280 X 200 pixels in High resolution monochrome mode.
Connector is phono socket.
Video RAM is $8000 to $BFFF on bank 1 (High resolution monochrome also uses $8000 to $BFFF on bank 2).

11. Screen 2 video output.
Screen 2 output is composite video monochrome, resolution is 640 X 200 pixels. Connector is phono socket. Video RAM is $8000 to $BFFF on bank 2. (this output is meaningless in color and high resolution monochrome modes).

12. TTL Color.
RGB at TTL levels plus separate TTL syncs are provided on a 6 pin DIN socket. (Compatible with BBC monitor plugs). 8 colors are possible in color mode. In monochrome modes this output gives the same picture as 'Screen 1 video'. Resolution is 640 X 200 pixels. (or 1280 X 200 in High resolution monochrome).
Bank 1 $8000 to $BFFF is luminance information.
Bank 2 $8000 to $BFFF is color information.
(in high resolution monochrome mode bank 1 holds even characters, bank 2 odd.

13. Analog Color.
RGB at 1V peak analog levels, with composite TTL level sync, are provided on a 7 pin DIN socket. 16 colors are possible in color mode. In monochrome modes this output gives the same picture as 'Screen 1 video'. Resolution is 640 X 200 pixels (or 1280 X 200 in High resolution monochrome).
Bank 1 $8000 to $BFFF is luminance information.
Bank 2 $8000 to $BFFF is color information.
(in High resolution monochrome mode bank 1 holds even characters, bank 2 odd).

14. Disk Drives.
Connectors for Sony 3.5 inch, Epsom 3.5 inch, and any 5.25 inch drive, are provided.
Single or Double density formats are software programmable.
Data rates of 125K, 250K, and 500K bits/second are possible.
Up to 4 drives can be connected at any one time.
Different drive types can be mixed, eg. 2 X Sony and 1 X 5.25 is allowed

Switch mode power supply unit 5amp at 5 Volt, 1.5amp at plus 12 and .5 amp at minus twelve.

MICROKEY Ltd.

Part of the A.T.S. Group

Languages available;
Microkey Forth-79 plus Fig Forth extensions. Assembler included. Operates on the 65xx family based systems, included in the package. Utilises discs under standard Forth-79 system.

Pascal, Cobol, Fortran, 'C', Fig Forth, and many other programs and languages are available using the Flex operating system which is available for the Microkey 4500 to bona-fide educational establishments only. (6809 based). At extra cost.

FORTH USER comment:

This Microkey FORTH computer originally was based on the REMSOFT specification for a low-cost FORTH micro, but soon outstripped that concept.
The PCN review is of the 4500 6502 version, which has the FORTH on board.
There is also a 6809 version (without FORTH on-board) with the FLEX operating system. This allows many other languages to be used.

Once sales of this powerful computer begin to rise, we hope to have software in Microkey-Forth available.

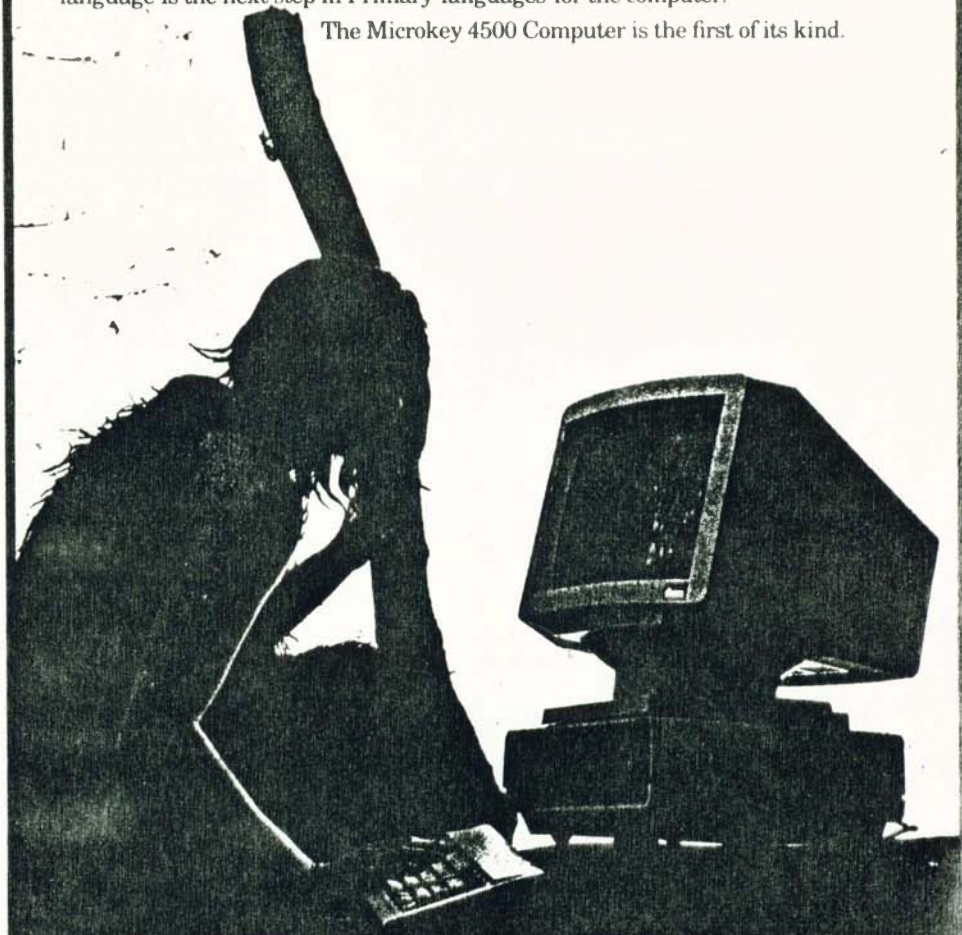
For further details of the computer, contact them at the address on their advertisement.

For details of possible software, contact John L Noyce, Publisher, PO Box 450, Brighton BN1 8GR.

EVOLVE

Mankind has evolved throughout the ages. His awareness, perception, intellect and speech have changed. So too have the extensions of these fundamental powers altered, from the flint axe to the computer. As we develop and adapt our computers, so too must the languages we use to program them change. The Forth language is the next step in Primary languages for the computer.

The Microkey 4500 Computer is the first of its kind.



System specification:

Forth 79 + Fig Forth extensions. Resident Assembler and Editor.
128K RAM in two 64K banks.
Video outputs.
Special Hi-res 1280 × 200 (minimum y value depends on monitor) Monochrome.
640 × 200 Colour matrix. 8 colours standard. 16 available as extra.
Dual monitor mode allows two independent 80 column monitors to be used, with separate scrolling. (Capability for dual-user with extra keyboard).
Universal RGB and sync outputs.
Character set is generated in RAM. (Foreign language character sets available including Arabic).
6502 (2 Mhz) Processor board with other processor capability on-board.
Full spec RS 232 port (TXD RXD RTS CTS DTR DSR DCD).
Integral switched mode PSU.
Full extender bus.
Metal chassis. Structural Foam case.
Cassette I/O.
Full QWERTY keyboard, on coiled cable, detachable.
3 parallel ports (bi-directional).
Disc Drives.
On board INCLUSIVE Disc controller for the Sony Microfloppies. Up to four Microfloppies may be used, two can be contained in the case. Each drive stores 300K (formatted) of 8-bit data.
Monitor commands in firmware.

Price £650 plus VAT. Sony Drive, installation kit. £210 per drive (up to 2 drives, consult us for 3-4 drives) plus VAT. Media £30 plus VAT for a pack of ten discs.

Sound monitor depicted. (Monitor is not included in the system but is available through us, recommended Medium Res monitor).

Information and orders to:
Microkey Ltd (Part of the ATS group)
98a St James St.
Brighton BN2 1TP.
Tel: (0273) 672911.

BOLDFIELD LIMITED COMPUTING

Sussex House, Hobson Street, Cambridge.
Tel: Ramsey (0487) 840740

THE

Jupiter ACE

IS BACK !

ONLY **£26** + VAT + p&p

WHILE STOCKS LAST

(REDUCED FROM £89.95)

THIS PRICE INCLUDES:

- THE JUPITER ACE COMPUTER WITH POWER SUPPLY
- 182 PAGE USERS MANUAL/FORTH PROGRAMMING BOOK
- DEMONSTRATION CASSETTE (5 PROGRAMS)
- TV & CASSETTE RECORDER LEADS
- SOFTWARE CATALOGUE
- FULL 12 MONTH GUARANTEE

ALSO:

16K RAM PACKS
JUST £20 + VAT
(Were £34.95)

JUPITER SOFTWARE
ONLY £3 + VAT
(Up to £7.95 previously)

* **SPECIAL DEAL** *

ACE + 16K FOR £44 + VAT + P&P
(Used to be £124.90)

The Jupiter Ace uses **FORTH** and features a full-size moving-key keyboard high resolution graphics, sound, floating point arithmetic and 3K of RAM.

ONLY BY MAIL ORDER

SEND S.A.E. FOR DETAILS